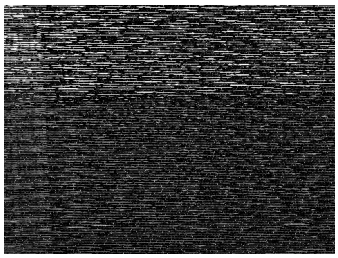


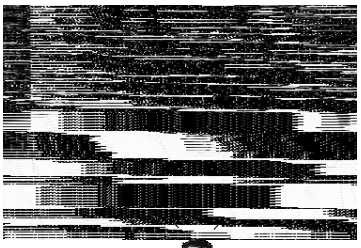
Motivation

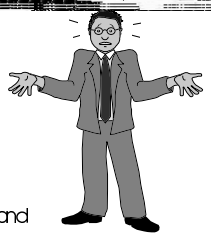
The needs for High Performance Computing

- High Performance Computing is important because we want to
  - Solve larger and larger scientific problems
    - Advanced Product Design
    - Environmental study, weather prediction
  - Store and process huge amount of data
    - Data Mining and Knowledge Discovery
    - Image processing, multi-media information
    - Internet information storage and search

Problem with Traditional Supercomputer

- Expensive
  - Very high starting cost
  - Expensive software
  - High maintenance cost
  - Costly to upgrade
- Vendor dependent





So, poor people can not do high performance computing.

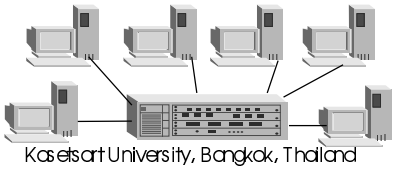
Solution Build a homemade supercomputer

Is it possible?

Yes!

PC Cluster: poor-man's supercomputer

- PC Cluster: A computing system built from high-end PCs and high-speed communication network.
- PC Cluster can be used as distributed memory multicomputer.
- PC Cluster support standard parallel programming based on message-passing model



Beowulf Project

- Originating from Center of Excelence and Information Systems Sciences (CESDIS) at NASA Goddard Space Center by Dr. Thomas Sterling, Donald Becker

  "Beowulf is a project ot produce the software for off-the-shelf clustered workstations based on commodity PC-class hardware, a high-bandwidth internal network, and the Linux operating system."

What is a Beowulf Cluster?

- Pid of PCs (POP)
- Pure M^2COTS (Mass Market Commodity Off The Shelves)
- Unix-like OS with source code availability: LINUX, BSD
- Message passing parallel programming model: MPI, PVM
- Scientific applications: Floating point intensive, data intensive
- computer science research and education

Father of Beowulf with Neagling Cluster at Caltech

Why Beowulf is good?

- Low initial implementation cost
  - M^2 COTS (Cheap PCs and Commodities Networks)
  - Free Software LINUX, GNU, MPI, PVM
- Scalability: computing system that grow and shrink with the needs
- Familiar Technology, easy for user to adopt the approach, use and maintain system.

Big Big Beowulf Cluster

- Avalon Cluster
- Large Beowulf Cluster (140 nodes), Los Alamos National Laboratory
- Dec Alpha Based Systems
- Deliver > 10 Gflops performance
- 88th fastest computer in the world

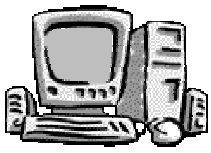
Our SMILE Beowulf Cluster

- 15 nodes
  - 4 PII 233 64MB
  - 8 PIII 350, 128 MB
  - 1 Dual Pentium II 450 MHz 128MB
  - 2 Frontend DUAL Pentium II 450MHz and DUAL Pentium Pro 200MHZ
- Storage around 50Gb
- Fast Ethernet Switch
- Peak performance about 1 Gflops

Cost consideration

What you need?

- A few good PCs
  - Pentium II 300-450
  - RAM at least 64Mb but 128-256 Mb is preferred
  - More than 2 Gb/Nodes
  - One Big monitor 17 inch for frontend
- High Speed Network



How much it cost you?

- PC price is now dropping fast
- Pentium II PC 400 MHz (Get from internet)
  - 6.4 GB-HD, CD 32X, RAM 64MB
  - 1249.95 $US
- Price per node will be about 1000-1500 US$
- About 80 Mflops per node
- Fast Ethernet 10/100 Switch
  - Low cost 250-450 US$
  - Brand Name 1000-2000 US$

So, let's play with some number

- Node cost = 1500 US$ (PII 400 MHZ)
- Front End cost = Node + Bigger Harddisk + Bigger and nicer Monitor = 2000 US$
- Fast Ethernet Switch is 500 US$ = 62 $/port
- Let's not count some cost of a while
  - Cabling, rack or shelf
  - UPS

Minimal 2 Nodes System

- Pentium 400, 100 Mbit Direct Connected Wire, no network required.
- Machine cost = 1500+2000 = 3500 US$ to get about 150 Mflops peak
- Dual CPU nodes get you about 300 Mflops peak for about 6000 US$

Our cost on 4 Nodes Systems (December 1997)

- Parts of our system (6904 US$)
  - 4 Pentium II 233 MHz
  - 2 GB EIDE, 64 Mb RAM, 2 FastEthernet Card per node
- Later, we purchased a fastEthernet hub for about 736 US$.
- Total is about **7640** US$ or 1910 US$/nodes including comm unication

Our new 8 nodes system (December 1998)

- Configuration
  - 8 Pentium II 350 RAM 128 HD 4.3GB
  - 2 FastEthernet switch (SMC TigerSwitch)
    - 1.8 GPbs Backplane, not blocking, cut through routing, SNMP support
- For 14736 US$ or 1842 US$/nodes including much better network
- Geting cheaper and better!

How much a 1Gflops peak performance system cost?

- Pentium II 400 is about 80 Mflops
- Need about 13 nodes (1500x12+2000x1 = 20000 US$)
- Need 16-24 port switch (2000 US$)
- Cost only about 22000 US$ or 22 US$/Mflops

Now, let's go shopping!

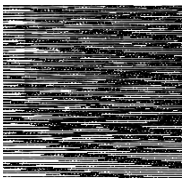
Choosing PC Cluster Components

Cluster Components

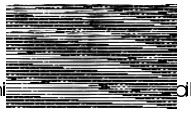
- Motherboard and case
- CPU and memory

- Hard Disk

- CD ROM, Floppy Disk
- Keyboard, monitor



- Interconnection network



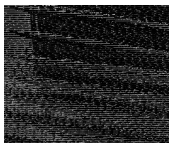
Mother Board

- Largest cache as possible (512 K at least)
- clock speed more than 100 MHz prefered
- Memory expansion
  - Normal board can go upto 512 Mbytes
  - Some server board can expand upto 1-2 Gbytes but very expensive (Intel, Tyan)
- Number and type of slots
  - at least 3-4 PCI slots is needed for SCSI Adaptor, Multiple network cards

MotherBoard

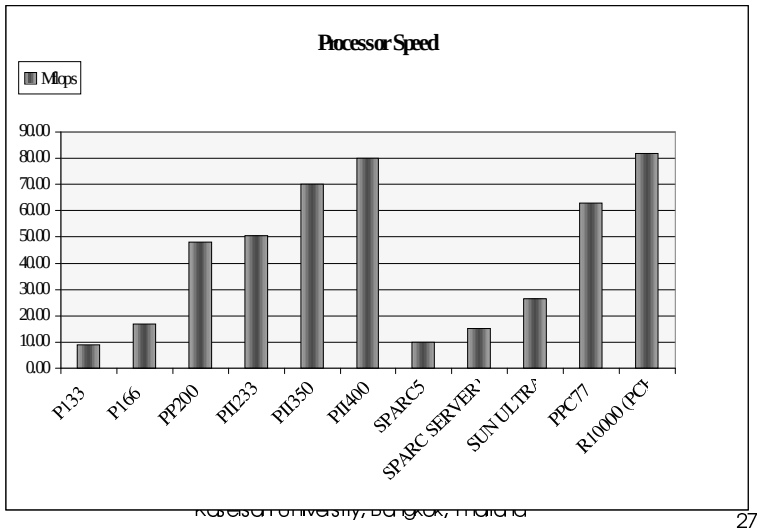
- Buid in option: SCSI, IDE, FLOPPY, SOUND USB
  - More reliable, less cost, but Inflexible
- Form factor
  - ATX case is typical now but AT is cheaper
- New motherboard usualy has buid in hardware monitor such as LM78. Highly recommended
- Compatbilty with Linux must be checked.

CPU



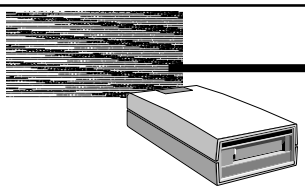
- Intel, CYRIX, 6x86, AMD are also supported
- Sometime avoid using newest generation processor can cut cost.
  - We found that for Intel Processor, 50Mhz increase in clock give you only about 10 Mflops more
  - Pentium 450 and Pentium 400 speed different is 10 Mflops (12-15%) but the price of 450 is almost twice that of 400.

Our linpack results on various CPU



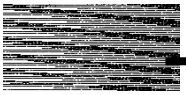
Memory

- SDRAM is better than EDO which is better than normal RAM
- SDRAM is now normal
- Pick fast SDRAM if possible. Access time as fast as possible: 60 nsec minimum
- SDRAM must match clock speed of main board (SDRAM 100 MHZ is needed for Mainboard 100 MHz)



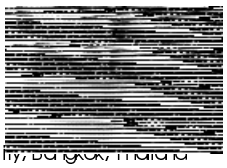
Hard Disk

- IDE : cheap and fast, controler buid-in on board, large capacity 10GB is avaiable.
- SCSI more durable, and can be expanded more than IDE (IDE is limited to 4 per node)
- IDE has a very short live for heavy, 24 hours use. Use SCSI if you need more reliable disk, IDE for low-cost solution.
- We use SCSI for common user space, IDE for scratch



CD-ROM, Floppy Disk Drive

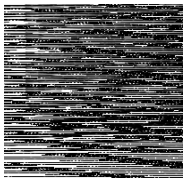
- CD-ROM will simplify the initial installation
- Need on one node only, possibly on front end
- Floppy is useful when booting or recovery from crash
- Need only 1.44 MB drive, don't need any fancy drive here.



Backup device

- Tape back up is necessary for long-term and reliable operation
- Linux support many tape and removable Devices
  - Normal Tape, slow but high capacity
  - IOMEGA Jazz Drive fast but low capacity (1GB)
- We use IOMEGA Jazz drive since we need a fast and random access capability.

Keyboard, Monitor



- Compute nodes, don't need keyboard and monitor (monitor, much lower cost)
- Front-end need big monitor for X windows, software development, demo.
- Need BIOS setup to disable keyboard on some system
- Keyboard and monitor switch is nice to have (but also cost you pretty good).

Interconnection Network

Many technologies are available

- ATM
  - Fast (155Mbps, 622Mbps)
  - Too expensive for this purpose
- Myrinet
  - Great, offer 1.2 Gigabit bandwidth
  - Still expensive
  - Very popular for high-end cluster

Interconnection Network

- Gigabit Ethernet:
  - Very fast
  - cost still quite high but will be cheaper soon
- Fast Ethernet
  - Cheapest a fast network available
- Most Beowulf cluster use fast ethernet, some high-end system use myrinet.

Fast Ethernet

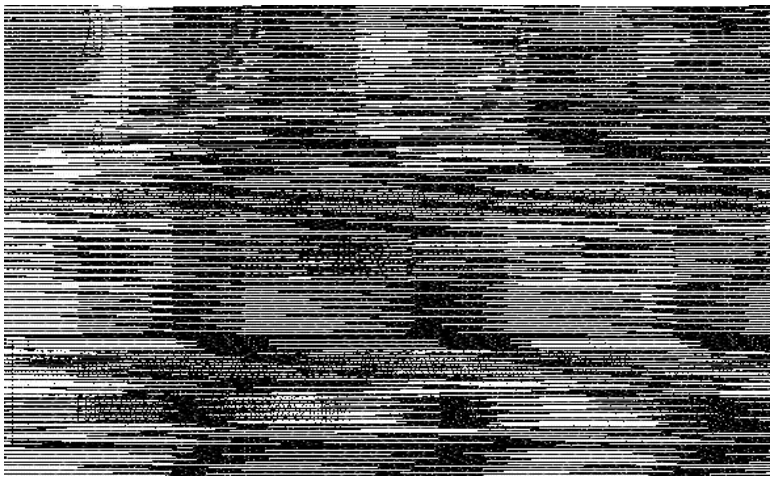
- The most popular network for cluster
- Getting cheaper and cheaper fast
- Offer good bandwidth
- Limit: TCP/IP Stack can pump only about 30-60 Mbps only
- Future technology: VIA (Virtual Interface Architecture) by Intel, Berkeley have just released VIA implementation on Myrinet

Network Interface Card

- For 100 Base-T, use only CAT-5 cable.
- NIC is buid in on some system.
- Availability of Linux Driver is important
  - Some card are not supported at all
  - Some supported card driver does not function properly.

Performance Comparison

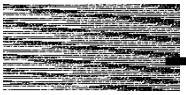
(from SCL Lab, Iowa State University)



System Planning

Planning for the Cluster

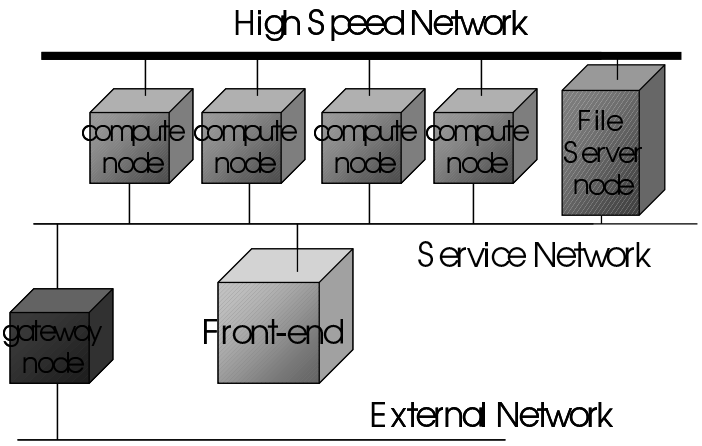
- System Configuration
- Software Configuration



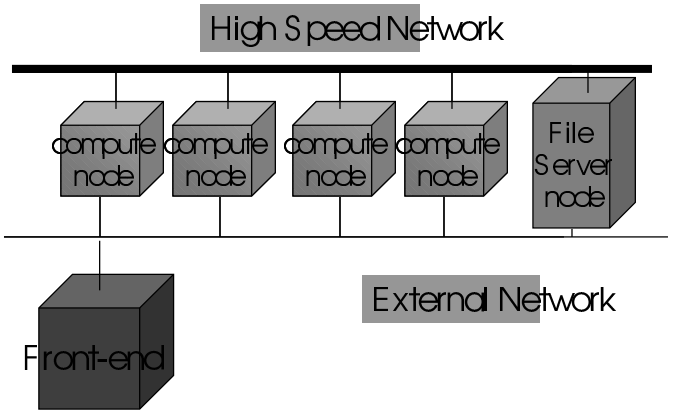
Type of Cluster

- Close Cluster: Hide most of the cluster behind gateway node
  - need less IP address, better security
  - Good for computing task
- Open Cluster: All nodes can be seen from outside
  - Need many IPs, more security concern
  - more flexible
  - Good for internet/web/information server task

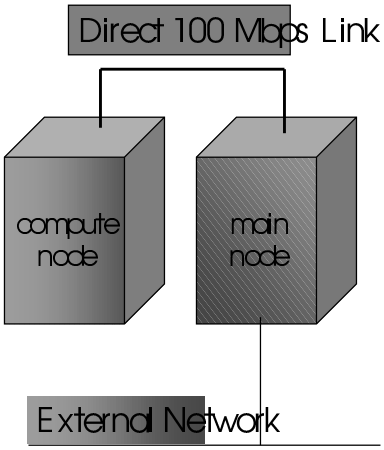
Close Cluster Configuration



Open Cluster Configuration



Minimal Configuration



- 2 Nodes
- Direct 100 Mbps link using crosswire
- Main node is also a file server

Interconnection Network

- Most popular one is Fast Ethernet
- Choices
  - Link node in to topologies such as mesh, Torus
    - Scalable, more bisection bandwidth
  - Switch Network
    - Easy, but more costly
  - Hub
    - lower-cost but lower performance

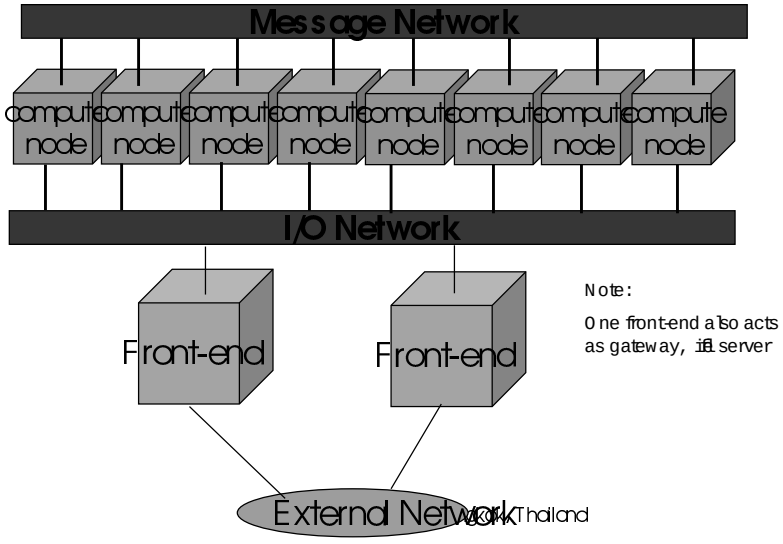
RJ-45 Wiring Diagram

If you hold RJ-45 connector facing you with the lock tab on the top from left to right. The pin usage is as follows

Pin Number	Assignment	Color
1	Output Data (+)	Blue/White
2	Output Data (-)	Blue
3	Input Data (+)	Orange/White
4	Reserved for Telephone	Orange
5	Reserved for Telephone	Green/White
6	Input Data (-)	Green
7	Reserved for Telephone	Brown/White
8	Reserved for Telephone	Brown

If you are connecting only two machines, this possible to avoid using a hub, by swapping the Rx and the Tx pairs (1-2 and 3-6)

Our SMILE Beowulf Configuration

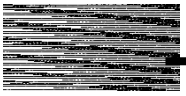


SMILE Hardware Configuration

Compute Nodes
4 Pentium II 233MHz
512 KB L2 Cache. (ASUS P2-L97 440LX), 64 MB, 2 GB EIDE,
2 Ethernet card on each node (3COM 3C905TX)
8 Pentium II 350MHz
128 MB , 4.3 GB EIDE, 440BX 100MHz board (ASUS)
2 Ethernet card on each node (3COM 3C905TX)
PII 450 DUAL Processors
4.3 GB IDE, 128 Mb, 100MHZ board

Front-end Nodes
PPRO200 DUAL Processors
Cache 512K 64 MB, 2 GB of EIDE hard disk, CD-ROM drive and 17 inches monitor,
Two Ethernet card, one is Fast Ethernet card and another is 10 Mbps Ethernet.
PII 450 DUAL Processors
10 GB IDE, 128 Mb, 100MHZ board

Network
100 Mbit Fast Ethernet Switch SMC

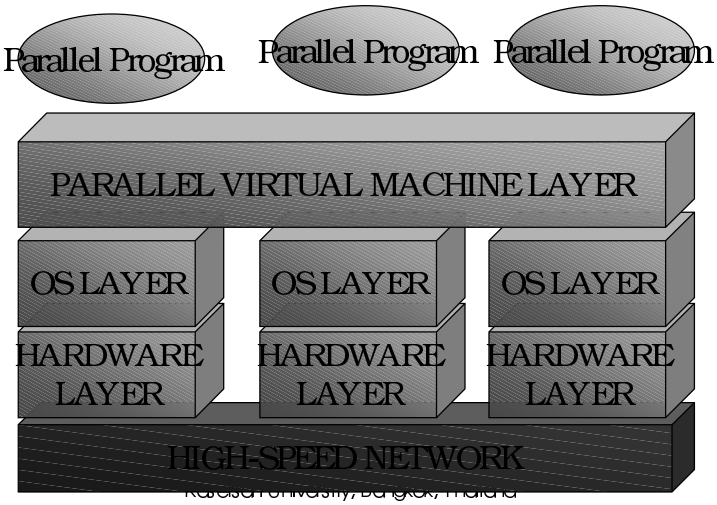


Software Configuration

Main software components
- Operating System
  - LINUX, FreeBSD
- Parallel programming System
  - PVM, MPI
- Utilities, Library, and public domain tools

These software are free!

Software Structure of PC Cluster



Single System View

- Single system view
  - Common filesystem structure view from any node
  - Common account on all nodes
  - Single software installation point
- Benefit
  - Easy to install and maintain system
  - Easy to use for users

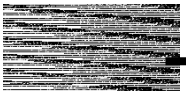
System Installation

Installation Steps

- Install Operating system (LINUX)
- Setup a Single System View
  - Common filesystem structure view from any node
  - Common account on all nodes, transparent remote login.
  - Single software installation point
- Install parallel programming system such as MPI, PVM, BSP
- Install utilities, libraries, and applications

Linux Installation

- Linux have many distribution: Slackware, Redhat, Caldera, Debian
- We use RedHat because
  - RedHat is easy to install, upgrade with RPM package management technology
  - RedHat come with very complete set of software
- Version used on SMILE is RedHat 5.1/5.2

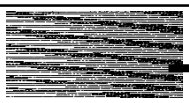


Quick Guide for Installation

- Planning the partition
  - Root filesystem (/)
  - Swap filesystems (twice the size of memory)
  - Shared directory on file server
    - /usr/local for global software installation
    - /home for user home directory on all nodes
- Planning IP, Netmask, Domain name, NIS domain

Basic Linux Installation

- Make bootdisk from CD or network distribution
- Partition harddisk according to the plan
- Select packages to install
  - Complete installation for Front-end, fileserver
  - Minimal installation on compute nodes
- Installation
- Setup network, X windows system, account

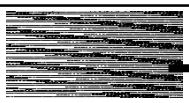


Basic Pitfall

- Linux hate plug-and-play. Turn it off using bios setup
- Set interrupt and DMA on each card to different interrupt ot avoid conflict. For plug-and-pain card, Windows can be used for this purpose.
- For front-end with 2 card, kernel must be recompiled to turn on IP masquerading and forwarding

Setup a Single System View

- Single file structure can be achieved using NFS
  - Easy and reliable
  - Scalability to large cluster is still questionable
- Autofs system can be used to reduce the load
  - mount file system when is used only
- We share /usr/local and /home from a single NFS server

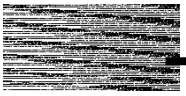


Centralized account

- Centralzied account using NIS (Network Information System)
  - Set NIS domain using domainname command
  - Start "ypserv" on NIS server u(sually file server of front-end)
  - run make in /var/yp
  - add "+:" at the end of /etc/password file and start "ypbind" on each nodes.
- Add all hodes in /etc/hosts.equiv to make transparent login.

MPI Installation

- MPI is a standard message passing interface for programming cluster or parallel system from MPI Forum
- MPI is easy since it is designed to be shared among many users
- Get MPI Distribution. Goo places to start are:
  - http://www.erc.msstate.edu/mpi/
  - http://www.mcs.anl.gov/mpi/mpich/

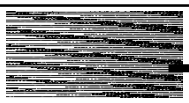


MPI Installation (MPICH)

- MPICH is a popular implementation by Argonne National Laboratory and Mississippy State University
- Installation (in /usr/local/mpi)
  - Unpack distribution
  - run configure
  - make
  - make prefix=/usr/local/mpi install
  - setup path and environment

PVM

- PVM Parallel Virtual Machines is another message passing from Oak Ridge National Laboratory and University of Tennessee Appear before MPI. Current version is PVM 3.4
- Flexible for non-dedicated cluster, easy to use
- Lower performance and less feature rich compared to MPI
- Information: http://www.epm.ornl.gov/pvm/



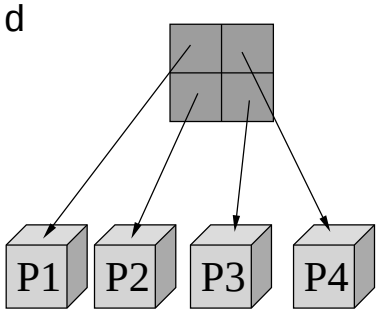
PVM Installation

- Get the distribution
- Unpack the distribution to use directory, PVM is difficult to install centraly
- Set environment
  - PVM_ROOT to pvm directory
  - PVM_ARCH to LINUX (or OS you use)
  - Set path to $PVM_ROOT/bin $PVM_ROOT/lib
- Go to pvm directory, run make file

Programming the Cluster

Power of Beowulf Comes from Parallel Processing

- Speeding up the execution by spliting task into many independent subtasks and run them on multiple processors concurrently
- Communication performance and task partitioning is a major concern



How to Program Beowulf Cluster

- Buiding applciations yourself using paralel message passing library such as PVM and MPI
- Using paralel lahguage such as High Performance Fortran and OpenMP
- Using publoic math library

Programming using Message Passing

- Partition task into multiple concurrent tasks that communicate by passing message
- MPI (message passing interface)
  - a de facto standard that is now supported by alplatform.
  - Free implementations are also avaiable (MPICH, LAM)

Programming using Message Passing

- Advantages
  - Standard and portable
  - High performance
- Disadvantage
  - Very difficult to program

MPI Program (1)

```
#include "mpi.h"
#include <stdio.h>
#include <math.h>
 double f(a)
double a;
{
    return (4.0 / (1.0 + a*a));
}
 int main(argc,argv)
int argc;
char *argv[];
{
    int done = 0, n, myid, numprocs, i;
    double PI25DT = 3.14159265358979323846264 3;
    double mypi, pi, h, sum, x;
    double startwtime, endwtime;
    int  namelen;
    char processor_name[MPI_MAX_PROCESSOR_NAME];
```

```
    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD,&myid);
    MPI_Get_processor_name(processor_name,&namelen);
     fprintf(stderr,"Process %d on %s\n",
            myid, processor_name);
     n = 0;
    while (!done)
    {
        if (myid == 0)
        {
/*
            printf("Enter the number of intervals: (0 quits) ");
            scanf("%d",&n);
*/
            if (n==0) n=100; else n=0;
             startwtime = MPI_Wtime();
        }
        MPI_Bcast(&n, 1, MPI_INT, 0, MPI_COMM_WORLD);
        if (n == 0)
            done = 1;
        else
```



```
  {          h   = 1.0 / (double) n;
           sum = 0.0;
              for (i = myid + 1; i <= n; i += numprocs) {
                  x = h * ((double)i - 0.5);
                  sum += f(x);
              }
             mypi = h * sum;
              MPI_Reduce(&mypi, &pi, 1, MPI_DOUBLE, MPI_SUM, 0,
        MPI_COMM_WORLD);
               if (myid == 0)
              {
                  printf("pi is approximately %.16f, Error is
  %.16f\n",
                         pi, fabs(pi - PI25DT));
                  endwtime = MPI_Wtime();
                  printf("wall clock time = %f\n",       endwtime-
  startwtime);
              }
          }
      }
      MPI_Finalize();
     return 0;
  }
```

Programming using Parallel Language

- There are hundreds of parallel programming language. But very few are standard.
- Widely used parallel language
  - HPF (High Performance Fortran)
    - Issued by HPF Forum,
    - SPMD data model, distributed memory
  - OpenMP
    - New standard by OpenMP Consortium
    - Supported by most commercial compiler
    - use shared memory model

Programming using Parallel Language

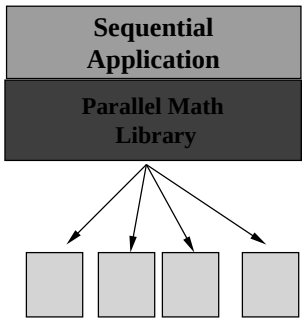
- Advantage
  - Easy to code, portable
- Disadvantage
  - Lower performance and scalability
- We are now using the Portland Group (PGI) HPF, F90, F77 compiler on our system

HPF Program

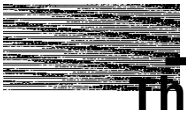
```
!
!       Parallel Fortran Program using HPF
!
!HPF$   PROCESSORS  p(number_of_processors())
        real c(6000,6000)
        real A(6000,6000) , B(6000,6000)
!HPF$   distribute  c(cyclic,*) onto p
!HPF$   align a(:,:) with c(:,:)
!HPF$   align b(:,:) with c(:,:)
        a(1:6000,1:6000)=10.0
        b(1:6000,1:6000)=5.0
        c = a * b
        print *,"Program is done"
        end
```


Programming using Parallel Math Library

- Many group try to hide the complexity of writing parallel code by providing easy to use math library
- Popular and free one are:
  - Scalapack by UTK
  - PetSc by Argonne National Lab
  - Plapack from UT Austin
- Advantage
  - Easy to code, no knowledge of parallel systems required



Application of Beowulf System



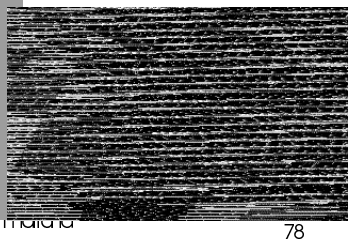
Things that you can use your Beowulf System s for

- HPC platform for scientific applications
  - This is the original purpose of beowulf project
- Storage and processing of large data
  - Satellite s image processing
  - Information Retrieval, Data Mining
- Scalable Internet/Intranet Server
- Main computing system in academic environment

Things that we use our beowulf cluster for

- Support courses in operating system, distributed systems, parallel computing
- Research
  - Scientific Computing
  - Parallel software tools
  - Parallel Information Retrieval
  - Parallel Search Engine

- Fluidized bed flow simulation
  - Study behavior of particles in fluidize-bed
  - Useful for chemical engineering industry
- Air pollution modeling
  - Simulate the movement of several substances in environment

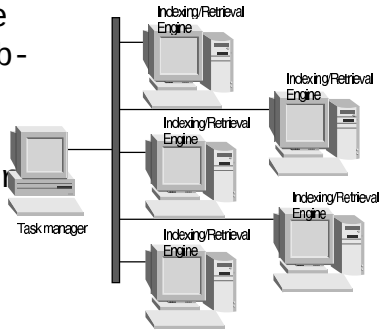


Parallel Software Tool Development

- Our system is used to support the development of Cluster Management System called SCMS
  - Parallel Unix Command
  - GUI Interface for Management and Performance Monitoring
  - Advanced Alarm System
  - Integrated Web Technology for system monitoring
- Availability: www.smile.cpe.ku.ac.th and beowulf project site (http://www.beowulf.org)

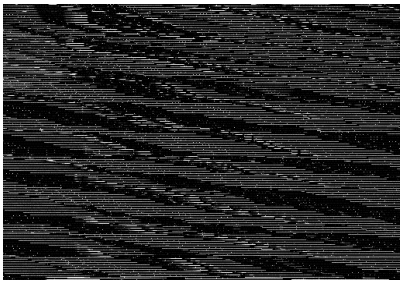
Distributed Text Retrieval on PC Cluster

- Approach
  - Sequential time-consuming and I/O bounded tasks are partitioned into several sub-tasks.
  - Distribute these tasks to multiple nodes of PC cluster to execute concurrently
- Collaboration with TREC project at NIST, USA.



Distributed Image Rendering

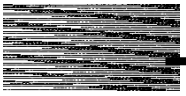
- Cluster similar to beowulf is used to render huge amount of image for movie "TITANIC"
- 160 DEC Alpha 433
- http://www.ssc.com/lj/issue46/2494.html



Conclusion

What we learn from doing this?

- PC Cluster is good platform for many application
- Limitations exist
  - Low speed, high latency network
  - Software inadequacy: tools and applications
- Software is important, not hardware
- Scalability and extendibility is a strong point



Clustered Computing

- Put cheap and powerful computing systems in the hand of scientist and engineer
- Save a lot of money
- Increase man power in scientific computing, parallel and distributed computing
- Provides an affordable platform for the research in Parallel/Distributed Computing, Computer Networking, Operating Systems etc..

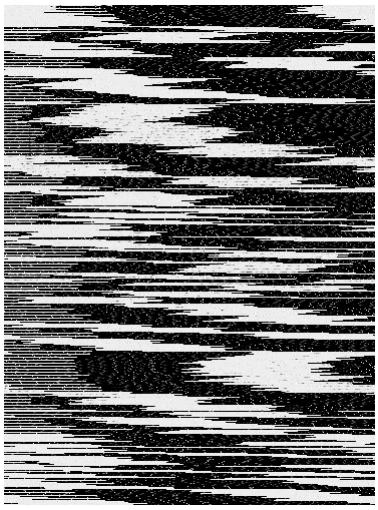
Future will not be the same after Beowulf

- Beowulf cluster will become the system most people used for HPC.
- Beowulf cluster give very high computing power to people.
  - People going to use it to do something we can not even imagine now
  - Same as PC revolution in 70s when Apple give cheap processing power to people.
- You ain't see nothing yet!

For more information

- Main beowulf page: www.beowulf.org
- Caltech Beowulf page
  http://www.cacr.caltech.edu/research/beowulf/
  - More detail tutorial on how to buid beowulf
- Cluster cookbook
  - Portland Group ClusterCookbook at
    http://www.pgroup.com/ClusterCookbook/
  - SCL cluster cookbook at Ames Research lab at
    http://www.sclameslab.gov/Projects/ClusterCookbook/

Brainwulf, an alternative approach to building a low cost parallel machine (from PARL, Clemson U.)



The End

Question?

For more information:
http://www.smile.cpe.ku.ac.th